Image Processing

Jim Emery

3/2/2011

Contents

1 Multiple Image Geometry

This geometry concerns the relationships between two or more two dimensional images of a three dimensional scene.

2 The Pinhole Camera

A pinhole camera consisting of an optical center C and a retinal plane R to which a 3d point M is projected by a ray passing through C and M. The image point m is the intersection of the ray with the retinal plane R. C of course does not lie on R.

3 A Camera With a System of Lenses

This camera can be treated almost like a pinhole camera. A system of lenses can be treated as a single lens, since each lens has a lens matrix, and the product of these matrices is a single matrix that may be used as if it were a single lens. Lenses are characterized by a set of points called the cardinal points of the lenses. Among these points are the two focal points and a pair of points called nodal points. A property of the nodal points is that a ray passing through the first nodal point exits from the second nodal point in a parallel direction. So this camera can almost be treated like a pin hole camera, where a ray entering the pinhole leaves the pinhole in a parallel direction.

4 Locating an Object Point When its Location in Two Stereo Images is Known

In this case we can trace a ray from each camera through the nodal points to the object point, which is the intersection of the two rays. Practically these two rays will not intersect exactly, so the shortest line between these two skew lines may be calculated and the image taken as the midpoint of this short line. Now as the lens system is focused the nodal points of the composite lens will change, so in order to compute these points information about the detailed arrangement of the lenses in the lens system is required. A lens system can be measured with a device called a nodal slide. However, in a camera the lens system may not be accessible. Apparently there are ways to characterize a camera without knowing it physical internal details by examining the images it produces.

5 The Epipoles

Consider a camera (C, R) and a second camera (C', R'), where C and C' are distinct. A line in 3d passing through C and C' meets the retinal plane R at a point $\mathbf{e}$, and meets the retinal plane R' at a point $\mathbf{e}'$. These are the epipoles.

6 Epipolar Lines

Given a 3d point M, the set of lines in the retinal plane R that pass through the epipole $\mathbf{e}$ and the image point m in R is called the epipolar line of m.

Similarly, for a second camera, the set of lines in the retinal plane R' that pass through the epipole $\mathbf{e}'$ and the image point m' in R' is called the epipolar line of m'. The image points m and m' are corresponding image points of the two cameras for 3d point M.

7 The Fundamental Matrix

The mapping taking image point m to the epipolar line ℓ'_m of m' is called the fundamental mapping. Its matrix $\mathbf{F}$ is called the fundamental matrix. So $\mathbf{F}$ is a 3 by 3 matrix. The fundamental matrix represents a mapping between the two dimensional projective plane of a first image to the two dimensional dual space (set of lines) of the second image.

8 The Epipolar Constraint

An image point m and its second image m' satisfy the epipolar constraint. To derive the constraint we have by definition that m' lies on its epipolar line. The epipolar line of m is obtained from the epipolar matrix $\mathbf{F}$ as

$$\ell'_m = \mathbf{F}m.$$

So the fact that m' lies on ℓ'_m means that in projective space

$$m' \cdot \ell'_m = 0.$$

That is,

$$\mathbf{m'}^T \mathbf{F} m = 0$$

This is the epipolar constraint between m and its second image point m'.

9 Camera Projection

The camera projection can be computed from a standard projection matrix with the camera pointing along the z axis and optical center at the origin. One uses a translation and rotation matrix to map to a coordinate system where the camera is in standard position.

10 An Example of Computing a Camera Projection Matrix: *camera.ftn*

11 Jerry Errett

```
Tuesday, February 15, 2011 8:49 PM
From:
"Jerry Errett" <jerry.errett@gmail.com>
To:
jdemery1@yahoo.com


Jim,

Thanks for talking to me about the robotics project at the cave
last Thursday. The math issue I have is finding the Fundamental
Matrix using multiple images. The goal is to construct a depth map
from two images. I have several books with the equations:
        The Geometry of Multiple images Faugeras and Luong
        Three-Dimensional Computer Vision  Faugeras
        Computer Vision and Applications  Jahne and Haubecker

The lisp version is SBCL  http://www.sbcl.org/

The project will be open source hardware and software.

I am looking forward to talking to you more this Thursday.

Thanks,
```

```
Jerry


Thursday, February 24, 2011 8:48 PM


Jim,
I was planning to go to the cave today but bailed due to the weather.
Thanks for the link to the FAB Lab info. I will try to make it out there Tuesday.

A use case for the image processing is:
      A robot has two forward facing color web cams placed 6 inches apart.
      It is on a brown table and the goal is to pick up a green tennis ball that is in front of it.
      The robot "snaps" a picture of the scene using both web cams.
      Using image recognition software the robot identifies the green ball in both images.
      Since the cameras are 6 inches apart the pixels containing the ball are offset in the two images.
      I want to use that offset to determine how far away the ball is from the robot.

      In practice there will a stream of video images from both cameras that need to be processed and the camera's can

I will try to be at the cave next Thursday.
Thanks for your help.

Jerry


On Thu, Feb 24, 2011 at 10:14 AM, jim emery <jdemery1@yahoo.com> wrote:

    Jerry,

    I have been reading the book "The Geometry of Multiple images" and doing a small
    amount of test programming. I can explain the Fundamental Matrix and the related
    projective geometry. I need to do a bit more reading to see the practicality of the use
    of this matrix. Also we might need to talk a bit more about what you want to accomplish with this image processing.

    With respect to machining there is a new FAB Lab at the Metropolitan Community College
    which has many machine tools and equipment  which may be made available to non-students. Make:KC is holding their '
    at this Lab from 6 to 9pm, and there will be a tour. For details:

    http://www.makekc.org/content/march-2011-show-tell
```

12 The Hough Transform

Note, this section is in the process of being partially updated, has been only partially changed, so needs further work, and needs to be checked, and so may be inconsistent, and incomplete

The Hough (pronounced "Huff") exploits the duality of projective space. Points in a projective space map to lines in the dual space, and also lines in the projective space map to points in the dual space. Let an image consist of points (pixels). Let the point in the plane have homogeneous coordinate vector (x, y, z). We can write the equation of a line containing this point as

$$xX + yY + xZ = 0.$$

The vector

$$\mathbf{p} = (x, y, z)^T$$

is the homogeneous coordinate vector of the point, and

$$\mathbf{L} = (X, Y, Z)^T$$

is the homogeneous coordinate vector of the line. So the equation of the line is given as an inner product

$$(\mathbf{p}, \mathbf{L}) = \mathbf{p}^T\mathbf{L} = 0.$$

If we define a line coordinate vector as

$$\mathbf{L_p} = \mathbf{p},$$

and a point coordinate vector as

$$\mathbf{p_L} = \mathbf{L},$$

then we have

$$(\mathbf{p_L}, \mathbf{L_p}) = 0.$$

So the mapping $\mathbf{p} \rightarrow \mathbf{L_p}$ is a mapping of a space of points into a space of lines. And under this mapping if a point $\mathbf{p}$ is contained in a line $\mathbf{L}$, then in the dual space, the dual line $\mathbf{Lp}$ contains the dual point $\mathbf{p_L}$.

So a set of points $\mathbf{p}_1, ..., \mathbf{p}_n$ are collinear, iff the lines in the dual space $\mathbf{Lp}_1, ..., \mathbf{Lp}_n$ are concurrent. Now suppose we map all of the points $\mathbf{p}$ of the image into the dual space and count the multiplicity of this mapping. That is for a given line $\mathbf{M}$ in the dual space, we count the number of times $k_\mathbf{M}$ that an image point has been mapped to this line. This tells us that $k_\mathbf{M}$ points lie on this line $\mathbf{M}$. If $k_\mathbf{M}$ is large then this line is probably in the image.

In order to deal with a finite region for the dual space, we normalize the homogeneous coordinates of the points to be of the form

$$ax + by = r,$$

where

$$a^2 + b^2 = 1,$$

and consequently there is an ω so that

$$a = \cos(\omega)$$

$$b = \sin(\omega).$$

A line is uniquely determined by the coordinates (ω, r). The Hough transform is a count of the number of image points that map to the line with coordinates (ω, r).

When the line is given in slope intercept form

$$y = mx + b,$$

$$y - mx = b$$

$$(-m, 1)^T \cdot (x, y)^T = b$$

Let

$$\mathbf{n} = \frac{(-m, 1)^T}{\sqrt{m^2 + 1}}$$

Then

$$\mathbf{n} \cdot (x, y)^T = \frac{b}{\sqrt{m^2 + 1}} = r,$$

where

$$\mathbf{n} = (\cos(\omega), \sin(\omega))^T,$$

for some ω. The standard equation of the line used in the Hough Transform is

$$x \cos(\omega) + y \sin(\omega) = r,$$

where

$$0 \leq \omega < \pi.$$

If $(x_1, y_1), (x_2, y_2)$ are two points on the line

$$\mathbf{n} \cdot [(x_1, y_1)^T - (x_2, y_2)^T] = [\frac{b}{\sqrt{m^2 + 1}} - \frac{b}{\sqrt{m^2 + 1}}] = 0.$$

So $\mathbf{n}$ is normal to the line. We have

$$\sin \omega = \frac{1}{\sqrt{m^2 + 1}},$$

$$\cos \omega = \frac{-m}{\sqrt{m^2 + 1}}.$$

If we discretize the (ω, r) space, we get a matrix, where each location in the matrix corresponds to a line of the dual space. The discrete Hough transform consists of this array, with the value in location (i, j) being the number of points in the original image whose dual lines have parameters $(\omega_i, r_i))$. Then a location with a high value is likely a straight line in the image. See **Hypercube Algorithms, with applications to Image Processing and Pattern Recognition**, by Sanjay Ranka and Sartaj Sahni, Springer-Verlag, 1990, p145.

See J R Parker **Practical Computer Vision Using C** page 175, for a Hough algorithm in C.

13 Bibliography

[1]Parker J R, **Practical Computer Vision Using C**, Wiley 1994 Joco Central, Hough Transform. Zerox copy.

[2]Ballard D H, **Generalizing The Hough Transform To Detect Arbitrary Shapes**, Pattern Recognition V13 N2 1981 pp111-122.

[3]Cowart A E, Snyder W E, Ruedger W H, **The Detection Of Unresolved Targets Using The Hough Transform**, Computer Vision, Graphics And Image Processing, V21 1983 pp222-238.

[4]Duda R O, Hart P E, **Use Of The Hough Transform To Detect Lines And Curves In Pictures**, Communication Of The ACM, V15 N1 1972 pp11-15.

[5]Ranka Sanjay, Sahni Sartaj **Hypercube Algorithms, with applications to Image Processing and Pattern Recognition** , Springer-Veralg, 1990.

[6]Dyer C R, **Guage Inspection Using the Hough Transforms**, IEEE Trans. On Analysis And Machine Intelligence, V PAMI-5 N6 1979 pp621-625.

[7] Olivier Faugeras and Quang-Tuan Luong **The Geometry of Multiple images**, MIT Press, 2001, Linda Hall.

[8]Faugeras **Three-Dimensional Computer Vision** ,

[9]Jahne and Haubecker **Computer Vision and Applications** ,

[10]Jahne Bernd **Digital Image Processing** Springer, 4th edition, 1997.